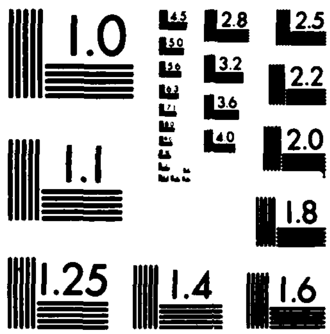
AD-A182 770 COMMON COMPUTER-BASEE TRAINING SYSTEM: A RECOMMENDED 1/1
APPROACH(U) AIR FORCE HUMAN RESOURCES LAB BROOKS AFB TX
T H KILLION ET AL. JUL 87 AFHRL-TP-86-61

UNCLASSIFIED F/G 5/9 NL

END
8-87
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FILE COPY

AFHRL-TP-86-61

(12)

AD-A182 770

AIR FORCE

HUMAN RESOURCES

COMMON COMPUTER-BASED TRAINING SYSTEM:
A RECOMMENDED APPROACH

Thomas H. Killion
Garry H. Boyle

OPERATIONS TRAINING DIVISION
Williams Air Force Base, Arizona 85240-6457

Barbara J. Eaton

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

July 1987

Interim Technical Paper for Period February - August 1986

LABORATORY

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

87 7 22 022

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

MICHAEL C. LANE, Colonel, USAF
Chief, Operations Training Division

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
A182 770

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFHRL-TP-86-61

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Operations Training Division

6b. OFFICE SYMBOL (If applicable)
AFHRL/OT

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Human Resources Laboratory
Williams Air Force Base, Arizona 85240-6457

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Air Force Human Resources Laboratory

8b. OFFICE SYMBOL (If applicable)
HQ AFHRL

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
Brooks Air Force Base, Texas 78235-5601

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO
62205F	1123	25	01

11. TITLE (Include Security Classification)
Common Computer-Based Training System: A Recommended Approach

12. PERSONAL AUTHOR(S)
Killion, T.H.; Boyle, G.H.; Eaton, B.J.

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM Feb 86 TO Aug 86

14. DATE OF REPORT (Year, Month, Day)
July 1987

15. PAGE COUNT
18

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	06	
05	08	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
computer-assisted instruction, (CAI) computer-managed instruction (CMI)
computer-based instruction, (CBI) computer proliferation,
~~computer-based training (CBT)~~ part-task training.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The rapid proliferation of microcomputer-based systems for uses ranging from administrative support through operations aids has led to interest in consolidation of requirements to facilitate acquisition, logistics, and maintenance. This paper considers the feasibility of a common system designed especially for training applications. This common computer-based training (CBT) system would be of modular design, thus making it adaptable to specific user requirements. An approach similar to the Z-150 acquisition is recommended, providing a means for speeding system acquisition without levying the requirement that it be the only system to be used. The advantages and disadvantages of such an approach are considered, along with recommendations concerning some of the capabilities and options to be included in its design.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Nancy J. Allin, Chief, STINFO Office

22b. TELEPHONE (Include Area Code)
(512) 536-3877

22c. OFFICE SYMBOL
AFHRL/TSR

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

AFHRL Technical Paper 86-61 July 1987

COMMON COMPUTER-BASED TRAINING SYSTEM: A RECOMMENDED APPROACH

Thomas H. Killion
Garry H. Boyle

OPERATIONS TRAINING DIVISION
Williams Air Force Base, Arizona 85240-6457

Barbara J. Eaton

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

Reviewed and submitted for publication by

Thomas H. Gray
Chief, Operational Unit Training Branch

SUMMARY

The rapid growth in the use of microcomputer-based systems for functions ranging from administrative support through operations aids has led to Air Force concerns about computer proliferation. A number of near-future acquisitions involve systems designed for training. As it now stands, these systems will each involve independent design, acquisition, logistics, and maintenance support. A possible solution to this problem is to consolidate requirements in order to produce a common computer-based training (CBT) system that could be adapted as needed to specific user requirements. This paper proposes an approach similar to that used in the Z-150 acquisition, thereby providing a vehicle for speeding acquisition and allowing widespread logistics support without legislating that it be the only system used. The advantages of such an approach include the cost benefits of a large buy, reduced development time for new applications, facilitation of logistics support, encouragement of industry standardization, and transportability of courseware/software across systems. Some of the core hardware/software capabilities to be incorporated into such a system include: (a) substantial computational capability and memory capacity, (b) advanced graphics, (c) videodisc interface, (d) sophisticated software development tools, (e) optional user interfaces, and (f) networking. Future considerations, such as artificial intelligence applications, enhanced authoring capabilities, and simulator interfaces are also discussed.

PREFACE

This effort represents a portion of the research and development (R&D) program of the Air Force Human Resources Laboratory (AFHRL) for Technical Planning Objective 3, the thrust of which is Aircrew Training. The general objective of this thrust is to identify and demonstrate cost effectiveness in training Air Force aircrew members. This paper was completed in response to a HQ USAF request to consider the design of a common computer-based system for aircrew training. It is being published with the hope of gaining a wider audience for consideration of these issues.

The authors would like to acknowledge the contributions of several individuals whose ideas and comments helped to refine the approach recommended herein. They are Lt Cols Bill Baltazar and Jim Coile, HQ USAF/XOOTW; Lt Col Mike Dickinson, AFHRL/ID; and Drs. Bernie Edwards and Bob Nullmeyer, AFHRL/OT.

TABLE OF CONTENTS

COMMON COMPUTER-BASED TRAINING SYSTEM:
A RECOMMENDED APPROACH

I. INTRODUCTION

Background

The Air Force is currently faced with the issue of proliferation of microcomputer-based systems. These systems are being acquired for a variety of applications, which run the gamut from administrative support through operational aids. Many of the near-future acquisitions involve systems designed for training. As it now stands, each of these systems will require independent design, acquisition, logistics, and maintenance support. For the individual operational squadron, this will result in a multiplicity of small computers, each with its own operating characteristics, interface requirements, and support network. Unless steps are taken to alleviate the situation, the individual squadron member will be required to have computer expertise on a variety of systems, and the squadron as a whole will have to cope with the maintenance problems associated with them. In addition to the problem of computer proliferation, there is pressure on the Air Force to develop a policy for computer-based training (CBT) systems. This pressure is the result of direction from Congress to the Department of Defense (DOD). If the Air Force does not develop such a policy, one may be imposed by fiat.

A possible approach to the issue of proliferation, particularly for CBT, is the concept of a common system. "Common" in this case implies the design of a system to fulfill multiple requirements, rather than to meet only a specific application. Such a system could be made available on a continuing contract, reducing response times for the development of new training applications that it can support. It should be stressed that such a system would not be levied as a requirement for all CBT applications. Its use would be determined by the particular training requirements involved. There are also some measures that could be taken to make other CBT application systems more compatible. Prior to a discussion of the proposed approach, a general philosophy concerning CBT will be discussed.

CBT Philosophy

As with the emergence of previous training technologies, CBT has been hailed by many as the "solution" to our training problems (Clark, 1983). As with sound/slide or videotape, initially there has been great enthusiasm, which will no doubt be followed by some disgruntlement concerning the difficulty of actually applying the technology. In fact, a considerable number of attempts to apply CBT to date have used it as a replacement for classroom or academic instruction. In such cases, the computer basically replaces the textbook as a source of information. This "page-turner" application of CBT takes minimum advantage of the capabilities that CBT provides. If the computer serves only as a replacement for a textbook, it is not a cost-effective solution. Instead, the instructional applications of CBT should take advantage of the unique capabilities of the computer itself, such as graphics, real-time interactivity, the flexibility to support multiple tasks, and the ability to support individually paced instruction.

Three areas where the computer provides unique opportunities are in real-time task simulation, performance measurement/scoring, and true computer-assisted/computer-managed instruction (CAI/CMI) (Edwards, in press). Real-time task simulation allows the student in initial training to get "on-task" as early as feasible in the instructional sequence, without the need for costly actual equipment trainers. In addition, this type of simulation has even greater applicability to operational training, where the trainee is already familiar with the basic task and simply requires practice in a variety of situations/conditions. Numerous studies have documented the relative equivalence of computer-based and actual equipment training (Babbitt, Pieper, Semple, & Swanson, 1985). Performance measurement/scoring serves two functions. First, it provides feedback to the student concerning task performance, allowing him/her to compare to a standard, observe improvements with practice, etc. Such feedback is essential if learning is to occur (Salmoni, Schmidt, & Walter, 1984). Well-designed feedback can be motivational as well as diagnostic. Second, performance measurement/scoring allows the system to adjust the level of the task to the capabilities of the student. This takes us into the realm of CAI/CMI, where the pacing of instruction and the types of cues/feedback provided to the student are driven by the student's capabilities. This is a depature from the lock-step nature of classroom instruction, where everyone must proceed at a common pace. Performance measurement/scoring also plays a role in CMI, where scheduling of training resources (including the computer) is based on student progress/performance.

Given these considerations, it is clear that any CBT system should provide some capability for task simulation and performance measurement. The complexity of the tasks simulated and the degree of CAI/CMI incorporated into the system drive the level of sophistication of the hardware/software required. At one end of the spectrum are devices that are primarily *page-turners*, with limited task simulation and requiring only limited graphics capability. At the other end of the spectrum are systems which are graphics-intensive, with the emphasis on complex task simulation.

The fact that such a broad spectrum of applications exists implies that any common CBT system will, in fact, need to be a family of systems. Some applications will not require the sophistication of a high-capability graphics system, and it would not be cost effective to use such a system in such cases. There should instead be some core hardware/software requirements which can serve as a building block for more or less sophisticated systems, depending on the training needs. The core requirements would ensure compatibility and transportability to some degree throughout the spectrum of systems. In this way, courseware could be adapted for the more simple applications or for more sophisticated systems as needed.

As should already be apparent, the types of systems being discussed here are microcomputer-based, with little or no operational hardware. This excludes "part-task trainers" such as the Air Refueling Part-Task Trainer or the Boom Operator Part-Task Trainer, which are actually crew station simulators for specific tasks. The systems under consideration here provide interactive training using computer-generated and/or videodisc displays, with student input via various interface devices (e.g., touch screen, joystick, mouse). In some cases, actual hardware may be interfaced with the system, such as an aircraft control stick, to increase the fidelity of interaction, but the majority of the system is not specific to a particular weapon system. This fact constitutes the basis for the possibility of a common CBT system. Such a system could support multiple training tasks via use of various software packages and (in some cases) individualized user interfaces. The system would need to be modular, as discussed previously, to adapt its sophistication to a specific application. However, such an approach is not well suited to existing acquisition policies, where only the minimum capability is acquired for each individual application. The possibilities offered by the Z-150 acquisition approach would appear to provide an attractive alternative. If an appropriately configured family of CBT systems could be designed, this approach would allow users to apply the system as needed to their training requirements without going through the full-scale development and acquisition process. This process now takes approximately 5 years, and leads to problems in acquiring state-of-the-art systems that are supportable within the rapidly changing computer industry.

Clearly, this common CBT system would need to be designed for training per se. It is essential to avoid having a system that is designed solely for other applications (e.g., data/word processing) designated as "the" training system. Although such systems are capable of performing many tasks, they do not necessarily have the features desirable in a training system. On the other hand, a system designed for training may have many of the features desirable in operational aids, and so it may be possible to use the same set of hardware for those applications.

II. APPROACH

Proposed Approach

The general approach recommended here, as mentioned above, is to use the Z-150 acquisition as a model for the development of a contract for a common CBT system, designed in advance to suit a variety of training needs. Based on the identification of training requirements, users would be able to order the appropriate number of systems, configured to meet their needs, through an existing contract. A maintenance contract would also exist to provide parts and service through a large-scale network. We believe this approach is feasible due to two factors. First, state-of-the-art microcomputers have reached the point where their capabilities rival those of minicomputers or even mainframe computers. This allows the use of the substantial memory and central processing unit (CPU) capacity required for supporting complex task simulations and significant CAI/CMI functions. Second, our experience with part-task trainer development and evaluation has suggested some of the capabilities we feel are necessary in such a common system, if it is to support a wide range of training tasks. Clearly, any such contract would have to incorporate provisions for advances in the state of the art, by building in growth potential and incorporating enhancements into newly delivered systems where it is cost effective to do so. The contract should also be limited to some prespecified period in recognition of the rapid advances in computer technology which improve the state of the art. The use of such a contractual approach has a number of advantages that will be discussed in the next section. However, there are two alternative approaches which deserve comment.

One alternative to the proposed approach would be to require everyone using CBT to use a core set of hardware. This would have the advantage of reducing logistics problems, while ensuring substantial compatibility and transportability of courseware. However, such an approach would require that all possible applications of CBT be identified in advance so that the appropriate capabilities could be designed in. This is nearly impossible to imagine. In addition, such an approach would lock the Air Force into a single manufacturer, with the attendant problems of production delays, sole-source buys, etc. It may also not be cost effective to always require the training developer to use a common system. In the case of total contracted training, it may be much more effective to allow the contractor to determine which type of CBT system best suits the overall design and is compatible with other facets of the training system. Given these and other considerations, we do not believe that this is a viable approach.

A second alternative would be to simply establish functional standards for all CBT systems, in an attempt to increase compatibility and transportability of courseware. Such standards might involve use of a 32-bit CPU to support complex applications, sufficient memory to take advantage of sophisticated software development tools, standard user interfaces, etc. Unfortunately, although we believe some standards are appropriate, it does not appear that this approach is very practical. The primary problem is in the graphics arena. There is simply no common graphics language in the industry. This means that graphics developed on one system would have to be converted to run on any other system. It may be wise for the Air Force to encourage industry standardization in this area; however, at least initially, selection of any particular graphics language would essentially imply sole-source hardware, with the attendant problems mentioned above.

Due to the problems with these alternatives, the proposed approach attempts a compromise between them. It does not require all CBT users to adapt to a single system but rather, provides them a vehicle for what we hope is a more responsive acquisition to meet their needs. It still provides the flexibility to use alternative systems, where cost or time or other considerations make such systems more appropriate. On the other hand, it goes beyond purely functional standards in that it assures compatibility and transportability for those who use the common system. It will also facilitate logistics support. This leads us into the advantages and disadvantages of such a common system, to which we now turn.

Advantages/Disadvantages

There are a number of advantages to the common CBT approach. First, there is the cost advantage of a large buy. Using the Z-150 acquisition approach as a model, a contract could be let to procure some minimum number of systems. This number would probably be quite large, considering the various applications across the Major Commands. Second, such an approach could reduce the development time for new training applications. In many cases, such development would primarily involve courseware. In those cases where new or additional hardware would be required, the existing contract would provide a vehicle for hardware acquisition, and courseware development could begin immediately on an existing system (if available). Third, having common hardware throughout the Air Force would reduce logistics problems. A single approach could be used for all systems, even to the extent of a single contract. This would increase the attractiveness of having support centers in multiple locations, including overseas, due to the large number of systems in the field. Fourth, the fact that the Air Force would have a widely used system for training would encourage industry standardization compatible with that system. With a large marketplace for hardware/software products for training, industry would no doubt develop software packages and hardware upgrades targeted for the system. One has only to consider the myriad of software and hardware products designed for the personal computer (PC) and Z-150 to understand the power of having a focus for industry development. One "area" where adaptation to a common CBT system would be particularly important would be authoring systems. Many existing authoring systems will run only on the hardware sold by the particular manufacturer. Having a common system would encourage manufacturers to adapt their authoring systems to this hardware, enabling them to sell courseware and courseware development capability in lieu of hardware. A further consideration is to design the common system with emphasis on projected industry standards to maximize the supportability of the system, as well as to take advantage of existing software products. Fifth, the use of common hardware/software would assure transportability of courseware across systems in a way that no other approach will. Again looking at the PC marketplace, despite claims of "PC compatibility," there are programs which will not run on PC "clones." In addition, even those programs that run are not necessarily as efficient on other hardware, due to minor differences in the architecture of the systems.

What are the disadvantages of a common hardware/software system? The primary problem is the dependence on a single manufacturer. This may be alleviated somewhat if industry responds by developing compatible systems. Any follow-on systems would clearly need to be compatible with existing software, to avoid conversion costs. This could also result in source problems, particularly in the volatile small-computer industry. A related problem would be the availability of systems due to production line constraints. Delays in the delivery of systems could have a domino effect on the development times for new training applications. Clearly, two of the criteria for selection of a manufacturer would have to focus on production capability and stability.

III. RECOMMENDATIONS

System Specifications

Considering the wide range of possible CBT applications, what kind of core system requirements can be established? The following are some recommended standards and options for such a system, along with a brief discussion of the basis for each item. There is no attempt to make an exhaustive listing; the intent is, rather, to provide representative considerations.

1. Computational Capability. The 32-bit CPU is fast becoming the industry standard. There will be Ada compilers for 32-bit systems, which will allow programs developed on this system to comply with the DOD standard. A 32-bit system also allows the downloading of existing mainframe computer programs that may have training applications. The system should obviously include an Ada compiler. Compilers for other languages (e.g., FORTRAN, C, Pascal) should be optional, to take advantage of existing programs.

2. Directly Addressable Memory. Combining the 32-bit CPU with at least 3 megabytes of random access memory (RAM) enables use of sophisticated software development tools (see para 10), as well as support for the real-time simulation of complex systems. The industry standard for RAM is already approaching 4 megabytes.

3. Graphics. Graphics capability is a keystone for the success of a common CBT system. Training of procedures or situational decision making require that the system present a recognizable situational display to the user. This may involve presentation of simulated instrumentation, a "God's-eye" view of a tactical environment, or even a three-dimensional perspective of an engagement, depending on the training objectives. To be effective, the system must present the critical cues for a given task in a format that is readily interpretable by the student. This depends on the graphics capability of the system, as well as the size and resolution of the monitor on which the graphics are displayed (see para 8). In addition, user acceptance and system effectiveness are directly related to rapid graphic drawing time in response to user inputs. This capability is essential in maintaining fidelity in the simulation of systems that have a rapid update rate.

There are several hardware and software features which would aid in achieving real-time graphics capability. Use of bit-mapped graphics with double buffering would be beneficial. With bit-mapped graphics, each picture element, or pixel, in a display has an associated memory word. This information can be transferred very rapidly in blocks to a display-list memory and subsequently to the system that drives the display signals. This can be compared to vector or stroke graphics, where only the endpoints of each vector are stored in memory. The latter requires less storage space in memory but imposes a heavy processing burden at display time, slowing graphics response time. Bit-mapped graphics require significantly more memory, but this capability is relatively cheap in today's systems. Double buffering essentially allows the system to have access to the data list for two displays simultaneously. While half of the list is being displayed, the other half is being rewritten with data for the subsequent screen display. This reduces access time and, consequently, system response time. It is particularly important in situations where animation is used. Another feature that facilitates graphics speed is the use of a dedicated subsystem for processing graphics data. In low-cost systems, this takes the form of a graphics co-processor. In systems supporting real-time, three-dimensional scenes, it is usually some sort of geometry engine. In both cases, its purpose is to remove the burden of graphics operations from the system CPU, allowing it to be devoted to such functions as overall program control, simulations, response monitoring, etc.

Given the key role of graphics in many training applications, program development time is an important consideration in the selection of a system. As mentioned earlier, there is no standard graphics language in industry, although attempts to develop standards such as the Graphic Kernel System (GKS) have begun. As a result, there are wide variations in the graphics libraries associated with different systems. Any system selected as the common CBT should have an extensive graphics library to facilitate rapid construction of simulated displays, scenes, and so on. This would benefit users employing the system's own graphics library, as well as simplifying the mapping between the system's graphics calls and the features available in a resident authoring system (see para 10).

4. Color. Color is a requirement in the majority of applications, in part due to the need to represent the actual systems faithfully. It may be convenient to simply include color as a requirement for all systems.

5. Sound. The design of the system should make provisions for inclusion of a sound generator to support presentation of the audio cues associated with particular systems. This is an area for possible growth in the domain of speech generation/recognition, for tasks involving non-scripted interaction with simulated participants (e.g., a controller). Possible sources of pre-recorded audio/speech are linear or still frame audio from a videodisc or digitized audio from a compact disc.

6. Interface Options. Provisions should be made for sufficient serial and parallel data ports to support interface with input/output devices such as a touch screen, joystick, mouse, printer, etc. The selection of devices will depend on the application. A keyboard should be standard.

7. Videodisc. Uses of interactive videodisc (IVD) for training are an ever-expanding domain. There are training situations where IVD is essential, but as with any new instructional tool, the developer can become enthralled with it and apply it inappropriately. Videodisc capability should be an option for the common CBT system, with its actual employment depending on the training requirement. The US Army has developed a standard for videodisc applications called the Electronic Information Delivery System (EIDS). The current design for EIDS employs the PC/AT architecture, with special computer boards added to handle graphics overlays on videodisc imagery and encoding/decoding of still frame audio and digital data (e.g., programs) on the videodisc. This design allows for use of any standard videodisc player, interfaced with these boards. If compatibility with EIDS is desired, either the common CBT system would have to accept these special-purpose boards, or new boards would have to be designed to be compatible with the system.

8. Monitor(s). The type of monitor(s) selected should be an option based on the specific training requirements. However, to maintain software transportability, some features, such as bit-mapping and color-fill capability, would have to be common across all systems. Monitors could vary in both size (e.g., 13", 19", or large screen for group lessons/debriefing) and resolution. Monitor size requirements would be a function of factors such as viewing distance, legibility, amount of material presented in a single frame, etc. A key factor in cases where a touch screen is used is to present controls of sufficient size to allow discrimination between touch points. Differences in resolution will require some modifications to the graphics to ensure proper display of identical material. In cases where high-resolution graphics and videodisc are both desirable, it may be advantageous to use two separate monitors, one high-resolution Red-Green-Blue (RGB) and one National Television Standards Code (NTSC). Current systems for displaying video resolution material on high-resolution monitors are costly. This may be an area where technological development will soon solve the problem.

9. User Interface. Use of an industry standard operating system, such as Unix, is recommended. A functional standard might also be implemented for a standard user interface. See "Standards for Other Systems" below.

10. Software Tools. Software and courseware are by far the most expensive portion of any training system. In order to reduce courseware development costs, software tools such as graphics development packages, programming languages, debuggers, and script writing tools should be made available. One approach to providing these tools in a format that does not require a programming background is to supply an authoring system. These systems are designed to allow a subject-matter expert (SME) to produce courseware, using a structured format. Morris, Braby, and Knight (1986) pointed out the advantages and disadvantages of such systems. In general, these

systems provide effective tools for efficient generation of certain types of courseware and require only limited training to use. However, they normally support a very small number of training formats (e.g., drill and practice) and have limited capabilities in the development of system simulations. Most likely, courseware will be developed through some combination of SME use of an authoring system and special-purpose programs developed by programmers. Most authoring systems do allow branching to programs developed outside the system (e.g., simulations). Such a combination would reduce the overall cost of courseware development without limiting applications to only those which can be developed within the context of the authoring system. As better authoring tools are developed, the amount of independent programming required should diminish.

In addition to courseware authoring capabilities, most of these systems support some degree of CMI. This includes such functions as scoring of student responses, record keeping, student advancement based on prior performance (adaptive instruction), scheduling, and report generation. These functions are critical for the training system manager in tracking student progress and managing training resources. The extent to which they can be automated directly affects the workload of the instructor(s). Any system proposed for use with the common CBT should be considered in terms of its CMI functions, as well as its authoring capabilities.

One example of a courseware authoring and CMI system is the Government-owned Instructional Support System (ISS). ISS was specifically designed to provide software transportability, in response to the computer proliferation problem. ISS is modular, enabling portions or the entire system to be loaded simultaneously. This feature allows ISS to run on a variety of equipment, ranging from microcomputers to mainframes. All editors in ISS are menu-driven and require no programming knowledge to operate. Minimal training is required in order to use the system. Three main editors comprise the authoring component of ISS: Graphics Editor, Simulation Editor, and Authoring Editor. This component allows the author to develop information displays, embedded questions, and individualized branching paths and combine them with graphics and simulations to produce courseware. The editors that comprise the CMI component enable the instructor to schedule resources and facilities, set up shifts and learning centers, and design curricula to meet individual needs. Prerequisite training, as well as acceleration/remediation of individual students, can be designed into a curriculum. ISS provides automatic assignment processing, processing and recording of student progress, data collection and analysis, and report generation. Many of the administrative tasks are taken care of, freeing the instructor to spend more time with students on a one-to-one basis. Improvements are clearly needed, such as the inclusion of a videodisc interface, enhanced tools for developing task simulations, and improved 2D and 3D graphics libraries. Some of these enhancements are already underway. ISS could be adapted to the common CBT system and provided as Government Furnished Equipment (GFE). As mentioned earlier, other courseware authoring houses might find it beneficial to adapt their systems to this hardware as well, in which case such systems could be selected at the discretion of the user.

11. Windowing. Windowing should be an option, allowing the system to continue to update "invisible" functions (e.g., displays) so that they can be called up immediately upon command. This also provides a means for decluttering the display, by presenting certain information only on demand and in a well-demarcated portion of the display.

12. Hard Disk. There is a wide range of options with respect to the capacity of a hard disk associated with a given system. Multiple options should be available, again depending on the training requirement and based on complexity of the programs, size of data bases (e.g., Defense Mapping Agency data), etc. In the case of Tempest-capable systems, the hard disk must be removable for storage. An alternative medium which is emerging as a high-density, large capacity data storage device is compact disc (CD). Currently, CD is a read-only memory (ROM) device, making it suitable for large-scale programs or data bases that do not require frequent updates.

13. Tempest. Tempest capability should be an option, again depending on the training requirement. Preferably, Tempest would be a characteristic of the system design, rather than simply building a "box" around the basic system. A Tempest system would have volatile memory and include a removable mass storage medium (removable hard disk, high-density floppy disk, or CDROM) small enough that it could be secured in a typical office safe. Consideration should be given to keeping the classified materials in files separate from the main program to simplify management of the material.

14. Networking. For some applications, it might be appropriate to network a number of individual delivery stations to a CPU, disk server, etc. This type of option would be most advantageous in a "schoolhouse" setting, where a high-capability, stand-alone system may not be necessary or even desirable. Most of the systems on the market have some provision for Ethernet or some equivalent. This should be a requirement for the system.

Standards for Other Systems

For those cases where the user decides not to use the common CBT system, we are still faced with the problem of training users to interact with each of the different systems. One approach to this particular problem would be to develop standards for the structure of the user interface, so that the user could expect to work with the same basic displays and commands in each case. This would include standards for structuring of menus, icons, definition of particular character strings, log-on procedures, etc. These standard formats would then lead the trainee into the more specific control structure of a given training program. This is an area for further study, to develop an optimal standard and consider the feasibility of its implementation.

Although the development of standards for the user interface would address the training issue, it would not solve the problem of courseware transportability across systems. A possible approach to this problem would be to develop a virtual machine interface (VMI). The VMI would be a set of standards for high-level calls to perform a given function. For example, in the area of graphics, there would be a standard call to place a circle on the display monitor in a certain position (e.g., by specifying a circle function with its associated attributes, such as a center point and a radius, filled with a particular color, etc.). Each hardware system would have specific interpreters in software/firmware to translate this call into its unique function(s) for producing the appropriate response. This would allow courseware from the common CBT system to run on other systems. However, it would require significant effort for each system to develop the machine-specific software/firmware to provide the VMI. In addition, this type of approach does not address the acquisition and logistics issues which a common contract does.

A final area for consideration of standards is in the area of authoring. As mentioned earlier, it is likely that authoring system vendors will find it advantageous to make their software compatible with the common CBT system. This would allow users to take advantage of a variety of authoring systems. However, this complicates matters when a particular user wishes to employ courseware developed on a "foreign" authoring system. There are several levels at which the user might wish to integrate this "imported" courseware into existing courses. At the least complex level, there would probably be little problem if all that was desired was to run a compiled lesson from this other system. The situation becomes more complex at a second level where the user wishes to tie this lesson into an existing CMI capability. This would require the imported courseware to provide standard outputs (e.g., percent correct, time on-task) in a recognizable format. This implies establishing standards for data streams which could be read by the CMI function of any system. The most complex level arises when the user wishes to incorporate portions of an imported lesson. To do this would require that the various authoring systems have common data formats for graphics, text, video control, etc. This would allow the host authoring system to read the imported courseware and pull out the necessary material. This would mean that there would have to be standardized data formats for text and graphics files, as there is now for videodisc imagery. This is clearly an area requiring further study, to determine the costs and benefits of achieving compatibility at these three levels.

Future Developments

One of the current problems in the development of CBT systems is the limited availability of expertise in the area of instructional design. In many cases, SMEs are used in the design of courseware. This is beneficial in terms of making sure that the content of the courseware is valid, but such personnel are usually unschooled in instructional techniques and approaches. Current authoring systems are generally designed to be "user friendly," but it takes some time to train SMEs how to use them. Even when the SMEs are trained, those systems that provide options in instructional format require some judgment on the part of the SME to determine which approach is most appropriate. There is also a tendency for the SME to become "enthralled with the tool" and to apply it in cases where other media may be more suitable. Since it is unlikely that the pool of instructional design personnel is likely to grow significantly, two options for addressing the problem appear viable. One approach is to use CBT itself to "train the trainer" in instructional techniques; that is, to use CBT to train SMEs in instructional design as well as in the use of a particular authoring system. This type of training would be a logical adjunct to existing authoring systems. The second approach is to develop more intelligent authoring tools, which require the SME to supply only the expert knowledge on which the courseware is based. The system itself would then "decide" what the proper format and sequence of instruction should be. Such systems would reduce development time and obviate the need for training SMEs in instructional design. Both of these options should be pursued. Use of CBT to train SMEs could provide a short-term solution while the data bases and systems are being developed to provide intelligent authoring in the long term.

The development of intelligent authoring tools is merely one aspect of the more general attempt to apply artificial intelligence (AI) techniques to CAI. Intelligent CAI (ICAI) is a rapidly expanding area, where tools such as intelligent tutors, expert and student models, and authoring aids are being investigated. If ICAI is to be successfully applied to systems such as the common CBT system, one of the keys will be the development of the necessary tools for this type of hardware. The majority of AI work is currently done on systems specially designed for symbolic processing using languages such as LISP and SNOBOL4. However, rapid developments within the microcomputer industry, in terms of computational capability and virtual memory, have made it feasible to do AI programming on low-cost hardware (Richardson, 1983). In addition, recent work has demonstrated the capabilities of Ada for use in AI programming (Reeker, Kreuter, & Wauchope, 1985). The latter development is particularly important for any common CBT system, as it would allow a single language to be used for the entire spectrum of applications. In this way, ICAI could be integrated with existing systems, rather than requiring separate, special-purpose hardware.

A final area of future concern is the interface between CBT and flight simulators. The issue here is the use of common software, or even hardware, to support system simulation for part-task and simulator training. There are ongoing programs investigating the possibility of "modularizing" simulators; that is, developing distinct modules to control particular portions of the simulation. Some of these modules, such as those for specific avionics systems, could be implemented on microcomputer systems operating in a distributed processing network. These modules would be relatively independent, to allow updates to individual programs without disrupting the overall simulation. The same programs used in these modules could be incorporated into the common CBT system to provide real-time task simulation for procedures training. The interface requirements would differ, but the basic task simulation would be identical. This would have the benefit of ensuring compatibility between the simulator and the part-task trainer. In addition, changes to the programs to reflect aircraft modifications would be simultaneously implemented on the two systems. Whether this commonality would extend into the hardware domain depends on engineering design considerations. The fact that future simulators will be designed to use Ada provides additional impetus to the use of Ada on the common CBT system.

IV. SUMMARY

This paper considers the feasibility of the concept of a common CBT system. We believe that an approach modeled after the Z-150 acquisition could make some inroads in solving the problems of computer proliferation, compatibility, transportability, etc. Clearly, any such system must be designed for training, although it may have applicability in other domains (e.g., operational aids). Some recommended capabilities have been presented, although the system should be modular so that the design can be adapted to specific training applications. This is not intended to be the final word but rather, a starting point for further discussion.

REFERENCES

Babbitt, B., Pieper, W., Semple, C., & Swanson, D. (1985). Effectiveness and efficiency of computer-based training: Recent research findings. In Proceedings of the Human Factors Society 29th Annual Meeting. Santa Monica, CA: Human Factors Society.

Clark, R.E. (1983). Reconsidering research on learning from media. Review of Educational Research, 53, 445-459.

Edwards, B.J. (in press). Advanced microcomputer-based aircrew part-task trainers. In R.F. Dillon (Ed.), Advances in testing and training (NPRDC-TR-86-9). San Diego, CA: Navy Personnel Research and Development Center.

Morris, C.L., Jr., Braby, R.T., & Knight, W.A. (1986). System design considerations for micro-computer based instructional laboratories (Technical Report 86-006). Orlando, FL: Naval Training Systems Center.

Reeker, L.H., Kreuter, J., & Wauchope, K. (1985, May). Artificial intelligence in Ada: Pattern-directed processing (AFHRL-TR-85-12, AD-A156 230). Lowry AFB, CO: Training Systems Division, Air Force Human Resources Laboratory.

Richardson, J.J. (1983, September). Artificial intelligence: An analysis of potential applications to training, performance measurement, and job performance aiding (AFHRL-TP-83-28, AD-A133 592). Lowry AFB, CO: Training Systems Division, Air Force Human Resources Laboratory.

Salmoni, A.W., Schmidt, R.A., & Walter, C.B. (1984). Knowledge of results and motor learning: A review and critical reappraisal. Psychological Bulletin, 95, 355-386.

END
8-87
DTIC